1. Student Details

This section should include your personal details, such as:

- **Name**: Rashi Sharma
- **Course/Program**: Modern Application Development II
- **University**: IIT Madras
- **Student ID**: 22f3000442

---

2. Project Details

Project Title: Influencer Engagement & Sponsorship Coordination Platform - V2

Project Question Statement:

This platform connects Sponsors and Influencers, enabling Sponsors to advertise their products/services while Influencers earn monetary benefits by promoting these products/services.

How the Problem Was Approached:

- **Problem**: Brands and companies often face difficulties in reaching the right influencers for their campaigns, and influencers struggle to find the right products/services to promote.
- **Solution**: The platform provides a seamless connection between Sponsors and Influencers, facilitating campaign creation, ad requests, negotiations, and payments. The admin can manage all users, campaigns, and statistics, ensuring a smooth experience for all parties involved.
- **Implementation**:
    - Used **Flask** for API development.
    - Implemented **SQLite** for data storage.
    - **Redis** was used for caching to improve performance.
    - **Vue.js** and **Bootstrap** were used for frontend development to ensure a responsive UI.
    - **Redis and Celery** were utilized for batch processing jobs like daily reminders and monthly activity reports.

---

3. Frameworks and Libraries Used

- **Backend**:
    - **Flask**: For creating the RESTful API.
    - **SQLite**: Lightweight database for storage.
    - **Redis**: Used for caching to optimize performance, especially for frequently accessed data.
    - **Celery**: For scheduling and running background tasks like daily reminders and monthly reports.
    - **Flask-Security/JWT**: For role-based authentication and user management.
- **Frontend**:
    - **Vue.js**: Used for the frontend, allowing for a dynamic and responsive UI.
    - **Bootstrap**: For styling and responsive layout.
    - **Vuex**: To manage the state, especially user authentication and data fetching.
    - **Axios**: For making HTTP requests to interact with the backend.
- **Additional Tools**:
    - **Flask-Mail**: For sending email notifications such as activity reports and reminders.

---

4. ER Diagram of Database
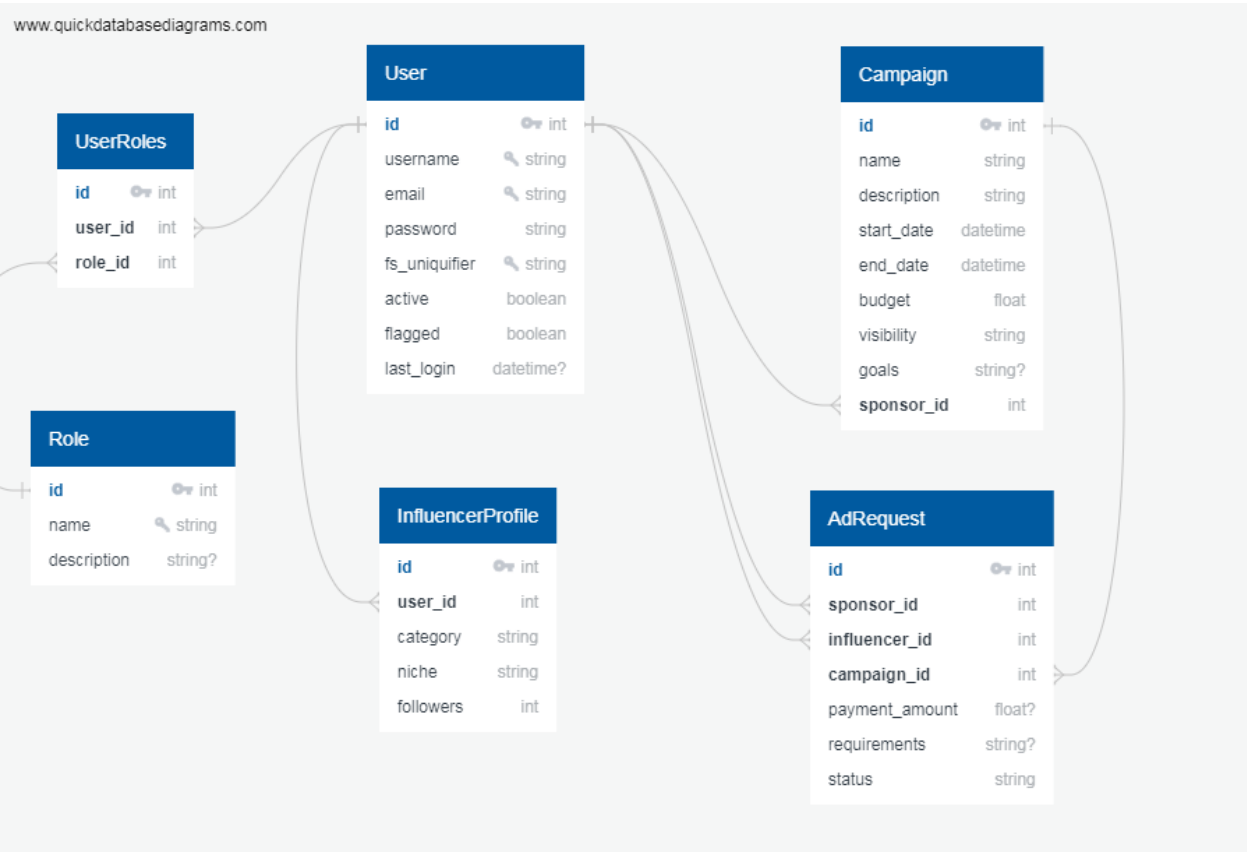
5. API Resource Endpoints

User Management

- **POST /api/login**: Login for sponsors, influencers, and admin.
- **POST /api/register**: Register new users (admin approval required for sponsors).
- **GET /api/user**: Fetch details of the logged-in user.

Admin Endpoints

- **GET /api/admin/dashboard**: Admin dashboard with all users..
- **POST /api/admin/flag_user/{user_id}**: Flag inappropriate users or campaigns.
- **GET /api/admin/stats**: admin stats on users, campaigns and requests.

Campaign Management (For Sponsors)

- **POST /api/campaigns**: Create a new campaign.
- **PUT /api/campaigns/{campaign_id}**: Update an existing campaign.
- **DELETE /api/campaigns/{campaign_id}**: Delete a campaign.

Ad Request Management (For Sponsors)

- **POST /api/ad_requests**: Create a new ad request for a campaign.
- **PUT /api/ad_requests/{ad_request_id}**: Update an existing ad request (accept, reject, negotiate).

Influencer Endpoints

- **GET /api/influencers**: List of influencers with optional search filtering.
- **GET /api/influencers/{influencer_id}**: Get details of a specific influencer.
- **POST /api/influencers/profile**: Create or update influencer profile information.

Other Functionality

- **POST /api/export_csv**: Export campaign details (public/private) as CSV for sponsors.

---

6. Drive Link of the Presentation Video

https://drive.google.com/file/d/1_e5MlsUysr_gSnnothj18HGuwYRRrZKx/view?usp=sharing

---